

WebPoint

Ein Framework zur Erstellung von webbasierten
Verkaufsanwendungen

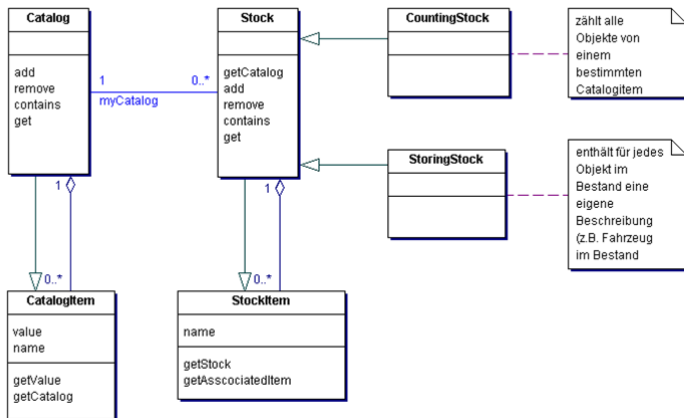
Maria C. Raabe
Daniel Woithe
Vorlage: Torsten Walter

16. Oktober 2006

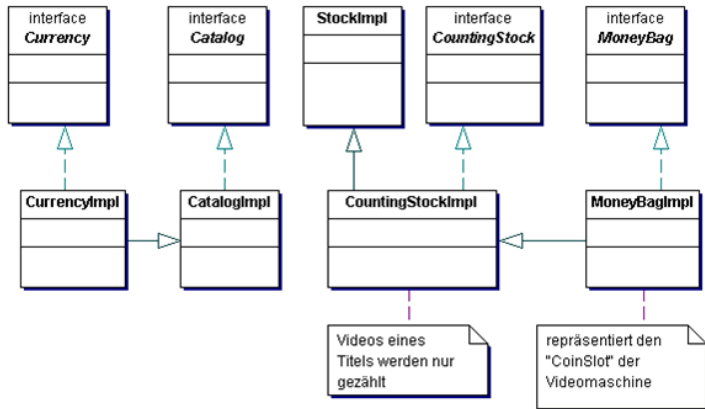
Gliederung

- 1 Datenverwaltung
- 2 Darstellung
 - Problemstellung
 - Java Server Pages
- 3 Interaktion
 - Links
 - Formulare

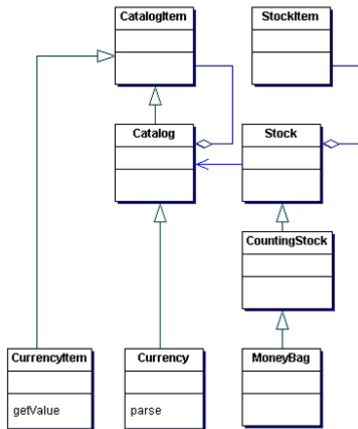
Kataloge & Bestände



Währungen & Geldbestände



Geldscheine & Münzen



Datenkörbe



Datenkörbe

- commit
- rollback



Datenkörbe

- commit
- rollback
- setCurrentSubBasket



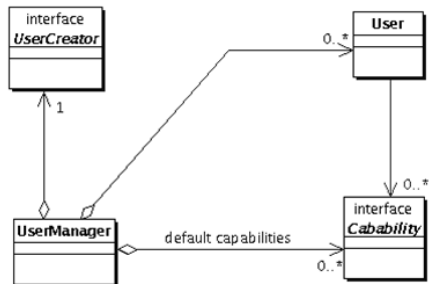
Datenkörbe

- commit
- rollback
- setCurrentSubBasket
- commitSubBasket
- rollbackSubBasket



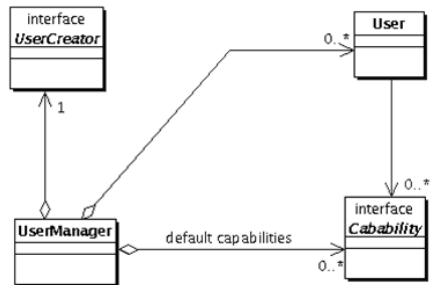
Benutzerverwaltung

- Benutzer



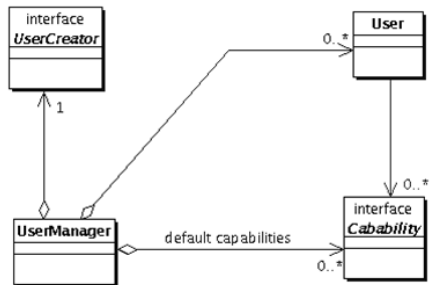
Benutzerverwaltung

- Benutzer
 - Namen
 - Passwort



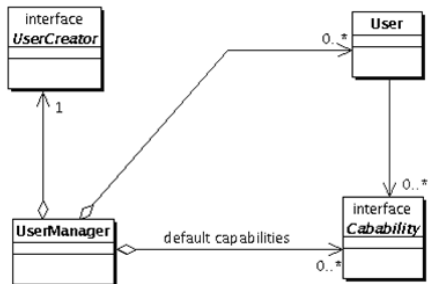
Benutzerverwaltung

- Benutzer
 - Namen
 - Passwort
 - Rechte



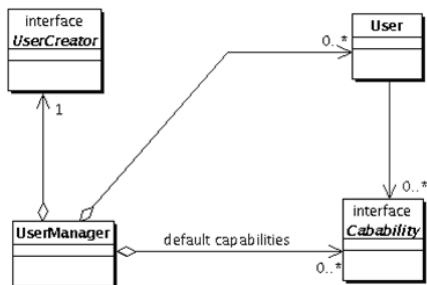
Benutzerverwaltung

- Benutzer
 - Namen
 - Passwort
 - Rechte
- Benutzerverwaltung



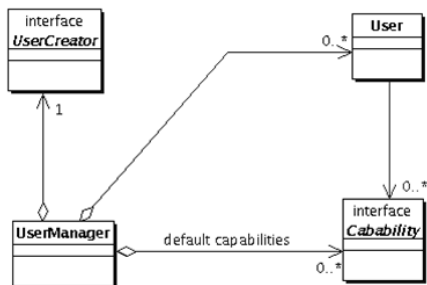
Benutzerverwaltung

- Benutzer
 - Namen
 - Passwort
 - Rechte
- Benutzerverwaltung
 - Benutzer hinzufügen
 - Benutzer löschen



Benutzerverwaltung

- Benutzer
 - Namen
 - Passwort
 - Rechte
- Benutzerverwaltung
 - Benutzer hinzufügen
 - Benutzer löschen
 - Benutzer erzeugen



Bis jetzt haben wir diverse **Java-Klassen** und **Interfaces** für die Verwaltung von:

- Katalogen
- Beständen
- Datenkörben
- Benutzern
- Protokollen

Was fehlt noch?

- Darstellung der Daten
- Interaktion mit dem Benutzer

Darstellung

Problem: Wir wollen die Benutzer mit Namen begrüßen.

Lösungsidee 1: Eine eigene Seite für jeden Benutzer.

- Hallo Anna
- Hallo Klaus
- Hallo Tom

Darstellung

Problem: Wir wollen die Benutzer mit Namen begrüßen.

Lösungsidee 1: Eine eigene Seite für jeden Benutzer.

- Hallo Anna
- Hallo Klaus
- Hallo Tom

Bedenken:

- unflexibel
- hoher Wartungsaufwand

Darstellung

Problem: Wir wollen die Benutzer mit Namen begrüßen.

Lösungsidee 2: Eine Seite als Template für alle Nutzer.

- Hallo %name%

Darstellung

Problem: Wir wollen die Benutzer mit Namen begrüßen.

Lösungsidee 2: Eine Seite als Template für alle Nutzer.

- Hallo %name%

Klingt besser, aber wie umsetzen?

Darstellung

Problem: Wir wollen die Benutzer mit Namen begrüßen.

Lösungsidee 2: Eine Seite als Template für alle Nutzer.

- Hallo %name%

Klingt besser, aber wie umsetzen?

Java Server Pages

- Hallo

```
<%= sc.getUser().getName() %>
```

Expressions

Beispiel1.jsp

```
<html>
<body>
  <h1>Hello World</h1>
  <%= new java.util.Date() %>
</body>
</html>
```

Expressions

Beispiel1.jsp

```
<html>
<body>
  <h1>Hello World</h1>
  <%= new java.util.Date() %>
</body>
</html>
```

Hello World

Sat Jul 23 14:23:33 CEST 2005

Import von Java Klassen

Beispiel2.jsp

```
<%@ page import="java.util.Date" %>
<html>
<body>
  <h1>Hello World</h1>
  <%= new Date() %>
</body>
</html>
```

Skriptlets

Beispiel3.jsp

```
<html>
<body>
  <% for (int i=0; i<10; i++) { %>
    <h1>Hello World</h1>
  <% } %>
</body>
</html>
```

Tag Library

Beispiel4.jsp

```
<%@ taglib uri="/tags/webpoint" prefix="wp" %>
<html>
<body>
  <% VideoShop shop =
      (VideoShop) Shop.getTheShop(); %>

  <wp:CountingStockTable
    cs = "<%= shop.getVideoStock() %>"
    ted= "<%= new TEDVideoStock() %>" />
</body>
</html>
```

Tag Library

Für die Darstellung von Tabellen gibt es die Tags:

- CatalogTable
- CountingStockTable
- StoringStockTable
- DataBasketTable
- LogTable

Name	Price	Count
Video-0	30,00 €	5
Video-1	30,00 €	5
Video-2	30,00 €	5
Video-3	30,00 €	5
Video-4	30,00 €	5
Video-5	30,00 €	5
Video-6	30,00 €	5
Video-7	30,00 €	5
Video-8	30,00 €	5
Video-9	30,00 €	5

Tag Library - Dokumentation

CountingStockTable - Display the contents of a CountingStock.

Render a HTML table that will display the contents of a `CountingStock`.

Attribute Name	Description
<code>cs</code>	The <code>CountingStock</code> whose contents are to be displayed. [Required] [RT Expr]
<code>db</code>	The <code>DataBasket</code> that is used to determine visibility of items. [RT Expr]
<code>cmp</code>	A <code>Comparator</code> defining the order in which to display the individual items. If <code>null</code> the records will be ordered by their keys. [RT Expr]
<code>showZeros</code>	If false, lines containing a '0' in the "Count" field will be hidden. [RT Expr]
<code>ted</code>	A <code>TableEntryDescriptor</code> that can split individual <code>CountingStockTableModel</code> records into a table's cells. Must not be <code>null</code> . [RT Expr]

Motivation

Bis jetzt können wir Daten

- **verwalten**
- **darstellen**

Jetzt wollen wir Nutzer Daten **manipulieren** lassen.

Was wird benötigt?

- JSP in der die Aktion aufgerufen wird
- Java Klasse mit der Aktion
- Konfiguration in transitions.xml
- Generierung der struts-config.xml

Beispiel: Zähler um eins erhöhen

JSP

```
<%@page import="vortrag.MainPlugIn" %>
<%@taglib uri="/tags/struts-html"
        prefix="html" %>
<html:html>
<body>
    Count: <%= MainPlugIn.count %>

    <html:link page="/addOneAction.do">
        add one
    </html:link>
</body>
</html:html>
```

Beispiel: Zähler um eins erhöhen

JSP

```
<%@page import="vortrag.MainPlugIn" %>
<%@taglib uri="/tags/struts-html"
         prefix="html" %>
<html:html>
<body>
  Count: <%= MainPlugIn.count %>

  <html:link page="/addOneAction.do">
    add one
  </html:link>
</body>
</html:html>
```

Beispiel: Zähler um eins erhöhen

JSP

```
<%@page import="vortrag.MainPlugIn" %>
<%@taglib uri="/tags/struts-html"
         prefix="html" %>
<html:html>
<body>
  Count: <%= MainPlugIn.count %>

  <html:link page="/addOneAction.do">
    add one
  </html:link>
</body>
</html:html>
```

Beispiel: Zähler um eins erhöhen

Java Action

```
package vortrag;  
public class AddOneAction extends Action {  
    public ActionForward execute (  
        ActionMapping mapping ,  
        ActionForm form,  
        HttpServletRequest request ,  
        HttpServletResponse response ) {  
        MainPlugIn.count++;  
        return mapping.findForward("ok");  
    }  
}
```

Beispiel: Zähler um eins erhöhen

Java Action

```
package vortrag;  
public class AddOneAction extends Action {  
    public ActionForward execute (  
        ActionMapping mapping ,  
        ActionForm form,  
        HttpServletRequest request ,  
        HttpServletResponse response ) {  
        MainPlugIn.count++;  
        return mapping.findForward("ok");  
    }  
}
```

Beispiel: Zähler um eins erhöhen

Java Action

```
package vortrag;  
public class AddOneAction extends Action {  
    public ActionForward execute (  
        ActionMapping mapping ,  
        ActionForm form,  
        HttpServletRequest request ,  
        HttpServletResponse response ) {  
        MainPlugIn.count++;  
        return mapping.findForward("ok");  
    }  
}
```

Beispiel: Zähler um eins erhöhen

Java Action

```
package vortrag;  
public class AddOneAction extends Action {  
    public ActionForward execute (  
        ActionMapping mapping ,  
        ActionForm form,  
        HttpServletRequest request ,  
        HttpServletResponse response ) {  
        MainPlugIn.count++;  
        return mapping.findForward("ok");  
    }  
}
```

Beispiel: Zähler um eins erhöhen

transitions.xml

```
<transition>  
  <page>/showCount</page>  
  <page action="/addOneAction" />  
  <page>/showCount</page>  
</transition>
```

Beispiel: Zähler um eins erhöhen

struts-config.xml (generiert)

```
<action-mappings>
  <action path="/addOneAction"
    type="***" >
    <forward name="***"
      path="/showCount.do"/>
  </action>
</action-mappings>
```

Beispiel: Zähler um eins erhöhen

struts-config.xml

```
<action-mappings>
  <action path="/addOneAction"
          type="vortrag.AddOneAction">

    <forward name="ok"
              path="/showCount.do"/>

  </action>
</action-mappings>
```

Beispiel: Zähler um eins erhöhen

struts-config.xml

```
<action-mappings>  
  <action path="/addOneAction"  
          type="vortrag.AddOneAction">  
  
    <forward name="ok"  
            path="/showCount.do"/>  
  
  </action>  
</action-mappings>
```



The screenshot shows a web browser window with a menu bar containing 'Datei', 'Bearbeiten', 'Ansicht', 'Gehe', 'Lesezeichen', 'Extras', and 'Hilfe'. Below the menu bar is a toolbar with icons for back, forward, refresh, stop, and home. The address bar displays the URL 'http://127.0.0.1:8080/vortrag/formular.jsp'. Below the address bar, there is a form with two input fields: 'Vorname:' containing 'Torsten' and 'Name:' containing 'Walter'. To the right of these fields is a button labeled 'absenden'.

Was passiert beim senden?



Was passiert beim senden?

```
http://127.0.0.1:8080/Vortrag/verarbeite.do?  
vorname=Torsten&name=Walter
```

Was wird benötigt?

- JSP mit dem Formular
- ActionForm
- Action
- Konfiguration in transitions.xml
- Generierung / Konfiguration in struts-config.xml

Beispiel: Vor- und Nachname eingeben

JSP

```
<%@ taglib uri="/tags/struts-html"  
        prefix="html" %>  
<html:html>  
<body>  
    <html:form action="/ExampleAction" >  
        Vorname: <html:text property="vorname"/>  
        Name:    <html:text property="name"/>  
        <html:submit> absenden </html:submit>  
    </html:form>  
</body>  
</html:html>
```

Beispiel: Vor- und Nachname eingeben

JSP

```
<%@ taglib uri="/tags/struts-html"  
        prefix="html" %>  
<html:html>  
<body>  
    <html:form action="/ExampleAction" >  
        Vorname: <html:text property="vorname"/>  
        Name:    <html:text property="name"/>  
        <html:submit> absenden </html:submit>  
    </html:form>  
</body>  
</html:html>
```

Beispiel: Vor- und Nachname eingeben

JSP

```
<%@ taglib uri="/tags/struts-html"  
    prefix="html" %>  
<html:html>  
<body>  
    <html:form action="/ExampleAction" >  
        Vorname: <html:text property="vorname"/>  
        Name:    <html:text property="name"/>  
        <html:submit> absenden </html:submit>  
    </html:form>  
</body>  
</html:html>
```

Beispiel: Vor- und Nachname eingeben

JSP

```
<%@ taglib uri="/tags/struts-html"  
        prefix="html" %>  
<html:html>  
<body>  
    <html:form action="/ExampleAction" >  
        Vorname: <html:text property="vorname"/>  
        Name:    <html:text property="name"/>  
        <html:submit> absenden </html:submit>  
    </html:form>  
</body>  
</html:html>
```

Beispiel: Vor- und Nachname eingeben

ActionForm

```
public class ExampleForm extends ActionForm {
    String vorname;
    String name;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    .
    .
}
```

Beispiel: Vor- und Nachname eingeben

ActionForm

```
public class ExampleForm extends ActionForm {
    String vorname;
    String name;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    .
    .
}
```

Beispiel: Vor- und Nachname eingeben

ActionForm

```
public class ExampleForm extends ActionForm {  
    String vorname;  
    String name;  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    .  
    .  
}
```

Beispiel: Vor- und Nachname eingeben

Action

```
package vortrag;
public class FormularAction extends Action {
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form ,
        HttpServletRequest request,
        HttpServletResponse response) {
        ExampleForm f = (ExampleForm) form;
        Person p = new Person(f.getName(),
                               f.getVorname());
        .
        return mapping.getInputForward();
    }
}
```

Beispiel: Vor- und Nachname eingeben

Action

```
package vortrag;  
public class FormularAction extends Action {  
    public ActionForward execute(  
        ActionMapping mapping,  
        ActionForm form ,  
        HttpServletRequest request,  
        HttpServletResponse response) {  
        ExampleForm f = (ExampleForm)form;  
        Person p = new Person(f.getName(),  
                               f.getVorname());  
        .  
        return mapping.getInputForward();  
    }  
}
```

Beispiel: Vor- und Nachname eingeben

transitions.xml

```
<transition>  
  <page>/formular</page>  
  <page action="/ExampleAction"  
        input="/formular" />  
</transition>
```

Beispiel: Vor- und Nachname eingeben

struts-config.xml (generiert)

```
<action-mappings>  
  <action path="/ExampleAction"  
    type="***"  
    name="***"  
    input="/formular.do" >  
  </action>  
</action-mappings>
```

Beispiel: Vor- und Nachname eingeben

struts-config.xml

```
<action-mappings>  
  <action path="/ExampleAction"  
          type="vortrag.FormularAction"  
          name="ExampleForm"  
          input="/formular.do">  
    </action>  
</action-mappings>
```

Beispiel: Vor- und Nachname eingeben

struts-config.xml

```
<form-beans>
  <form-bean name="ExampleForm"
             type="vortrag.ExampleForm"/>
</form-beans>

<action-mappings>
  <action path="/ExampleAction"
         type="vortrag.FormularAction"
         name="ExampleForm"
         input="/formular.do">
  </action>
</action-mappings>
```

Beispiel: Vor- und Nachname eingeben

struts-config.xml

```
<form-beans>  
  <form-bean name="ExampleForm"  
             type="vortrag.ExampleForm"/>  
</form-beans>
```

```
<action-mappings>  
  <action path="/ExampleAction"  
          type="vortrag.FormularAction"  
          name="ExampleForm"  
          input="/formular.do">  
  </action>  
</action-mappings>
```

Fragen?